10. Problem Statement: Machine Scheduling

**Problem Analysis:**

You are given n tasks and an infinite supply of machines on which these tasks can be performed. Each task has a start time si and a finish time fi, si< fi. [ si, fi ] is the processing interval for task i. Two tasks i and j overlap iff their processing intervals overlap at a point other than the interval start or end. For example, the interval [1, 4] overlaps with [2, 4] but not with [4, 7]. A feasible task-to-machine assignment is an assignment in which no machine is assigned two overlapping tasks. Therefore, in a feasible assignment each machine works on at most one task at any time. An optimal solution assignment is a feasible assignment that utilizes the fewest number of machines. A greedy way to obtain an optimal task assignment is to assign the tasks in stage, one task per stage and in non-decreasing order of task start times. Call a machine old if at least one task has been assigned to it. If a machine is not old, it is new. For machine selection, use the greedy criterion: If an old machine becomes available by the start time of the task to be assigned, assign the task to this machine; if not, assign it to a new machine.

**Algorithm:**

for i:= 1 to n do

{

for j:= i to n do

{

if (finish [i] > finish [j])

{

swap (task [i], task [j]);

swap (start [i], start [j]);

swap (finish [i], finish [j]);

}

}

}

for i := 1 to n do

{

for j:= i to n do

{

if (finish [i] == finish [j])

{

if (start [i] > start [j])

{

swap (task [i], task [j]);

swap (start [i], start [j]);

swap (finish [i], finish [j]);

}

}

}

}

for i := 1 to x do m[i] = 0;

for i := 1 to n do state[i] = 0;

for i := 1 to x do

{

mfns = 0;

for j := 1 to n do

{

if (state [j] == 0 and mfns<=start[j])

{

machine[i][m[i]++] = task[j];

state[j] = 1;

mfns = finish[j];

cnt++;

}

}

}

**Source Code:**

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int n,x;

printf("Please enter the number of task and number of machines:\n");

cin>>n>>x;

int task[n+1];

cout<<"Please enter the starting time for "<<n<<" works:\n";

int start[n+1];

for(int i = 0; i < n; i++) task[i] = i;

for(int i = 0; i < n; i++) cin>>start[i];

int finish[n+1];

cout<<"Please enter the finishing time for "<<n<<" works:\n";

for(int i = 0; i < n; i++) cin>>finish[i];

for(int i = 0; i < n; i++)

{

for(int j = i+1; j < n; j++)

{

if(finish[i]>finish[j])

{

swap(task[i],task[j]);

swap(start[i],start[j]);

swap(finish[i],finish[j]);

}

}

}

for(int i = 0; i < n; i++)

{

for(int j = i+1; j < n; j++)

{

if(finish[i]==finish[j])

{

if(start[i]>start[j])

{

swap(task[i],task[j]);

swap(start[i],start[j]);

swap(finish[i],finish[j]);

}

}

}

}

int machine[x][n], m[x], state[n], cnt = 0, mfns;

for(int i = 0; i < x; i++) m[i] = 0;

for(int i = 0; i < n; i++) state[i] = 0;

for(int i = 0; i < x; i++)

{

mfns = 0;

for(int j = 0; j < n; j++)

{

if(state[j]==0&&mfns<=start[j])

{

machine[i][m[i]++] = task[j];

state[j] = 1;

mfns = finish[j];

cnt++;

}

}

if(cnt==n) break;

}

cout<<"Solution:\n";

for(int i = 0; i < x; i++)

{

cout<<"Machine "<<i+1<<": ";

for(int j = 0; j < m[i]; j++)

printf("%c ",machine[i][j]+'a');

printf("\n");

}

return 0;

}

**Sample Input:**

Please enter the number of task and number of machines:

7 3

Please enter the starting time for 7 works:

0 3 4 9 7 1 6

Please enter the finishing time for 7 works:

2 7 7 11 10 5 8

**Sample Output:**

Solution:

Machine 1: a b e

Machine 2: f g d

Machine 3: c